-- Alexa Schnetzler Project 2

--Part 1 Question 1

--Add Notes Column to mm\_rental

alter table mm\_rental add notes varchar(255);

--Procedure

--Procedure

CREATE OR REPLACE PROCEDURE MOVIE\_RENT\_SP

(p\_memId in OUT NUMBER,

p\_movieId in OUT Number,

p\_pay in OUT Number)

AS

lv\_qty NUMBER(5,2) := 1;

lv\_stock NUMBER(5,2);

no\_info EXCEPTION;

PRAGMA EXCEPTION\_INIT(no\_info, -02291);

BEGIN

--Create new order

Insert into mm\_rental(rental\_id,member\_id, movie\_ID, PAYMENT\_METHODS\_ID)

VALUES (MM\_RENTAL\_SEQ.NEXTVAL,p\_memid, p\_movieId, p\_pay);

--Update Stock

Update mm\_movie

set MOVIE\_QTY = movie\_QTY - lv\_qty

WHERE p\_movieId = movie\_ID

AND movie\_qty > 0;

--Put stock into variable to output

select MOVIE\_QTY

into lv\_stock

from mm\_movie

WHERE p\_movieId = movie\_ID

AND movie\_qty > 0;

--Output everything

DBMS\_OUTPUT.PUT\_LINE('New Order Report Created On: ' || sysdate);

DBMS\_OUTPUT.PUT\_LINE('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

DBMS\_OUTPUT.PUT\_LINE('Rental Id: ' || MM\_RENTAL\_SEQ.NEXTVAL);

DBMS\_OUTPUT.PUT\_LINE('Member Id: ' || p\_memId);

DBMS\_OUTPUT.PUT\_LINE('Movie Id: ' || p\_movieId);

DBMS\_OUTPUT.PUT\_LINE('Payment Type: ' || p\_pay);

DBMS\_OUTPUT.PUT\_LINE('Movie Stock: ' || lv\_stock);

--Exceptions

exception

when no\_info then

DBMS\_OUTPUT.PUT\_LINE('One of your values does not exist in our database. Please review your values and try again.');

when NO\_DATA\_FOUND then

--Update Stock

Insert into mm\_rental(rental\_id,member\_id, movie\_ID, CHECKOUT\_DATE, PAYMENT\_METHODS\_ID, notes)

VALUES (MM\_RENTAL\_SEQ.NEXTVAL,p\_memid, p\_movieId, NULL, NULL, 'Order not completed. Movie not in stock. Contact customer.');

DBMS\_OUTPUT.PUT\_LINE('Sorry, that movie is currently not in stock.');

END MOVIE\_RENT\_SP;

--Anon Block SUCCESSFUL

Declare

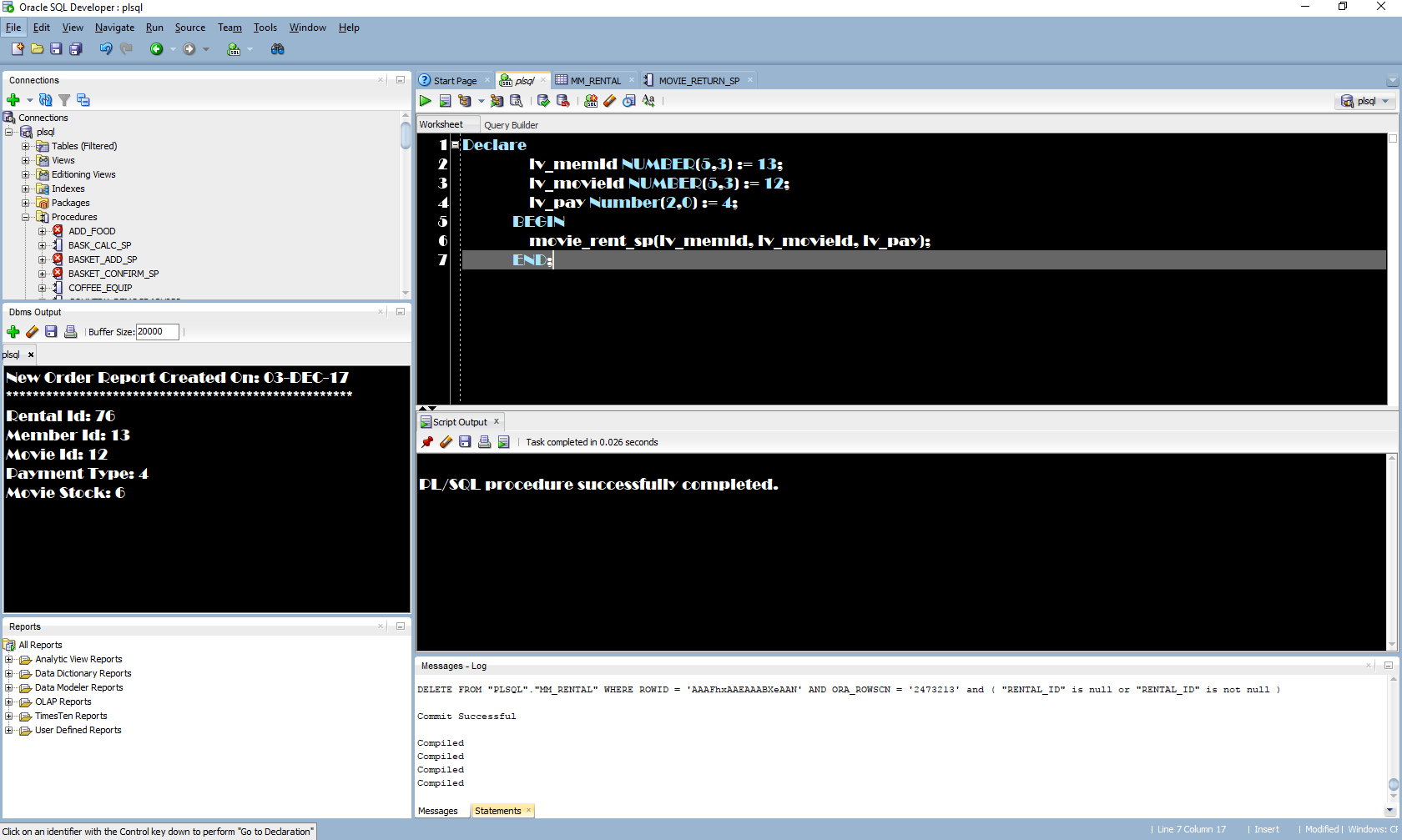
lv\_memId NUMBER(5,3) := 13;

lv\_movieId NUMBER(5,3) := 12;

lv\_pay Number(2,0) := 4;

BEGIN

movie\_rent\_sp(lv\_memId, lv\_movieId, lv\_pay);

 END;

--Anon Block Error

Declare

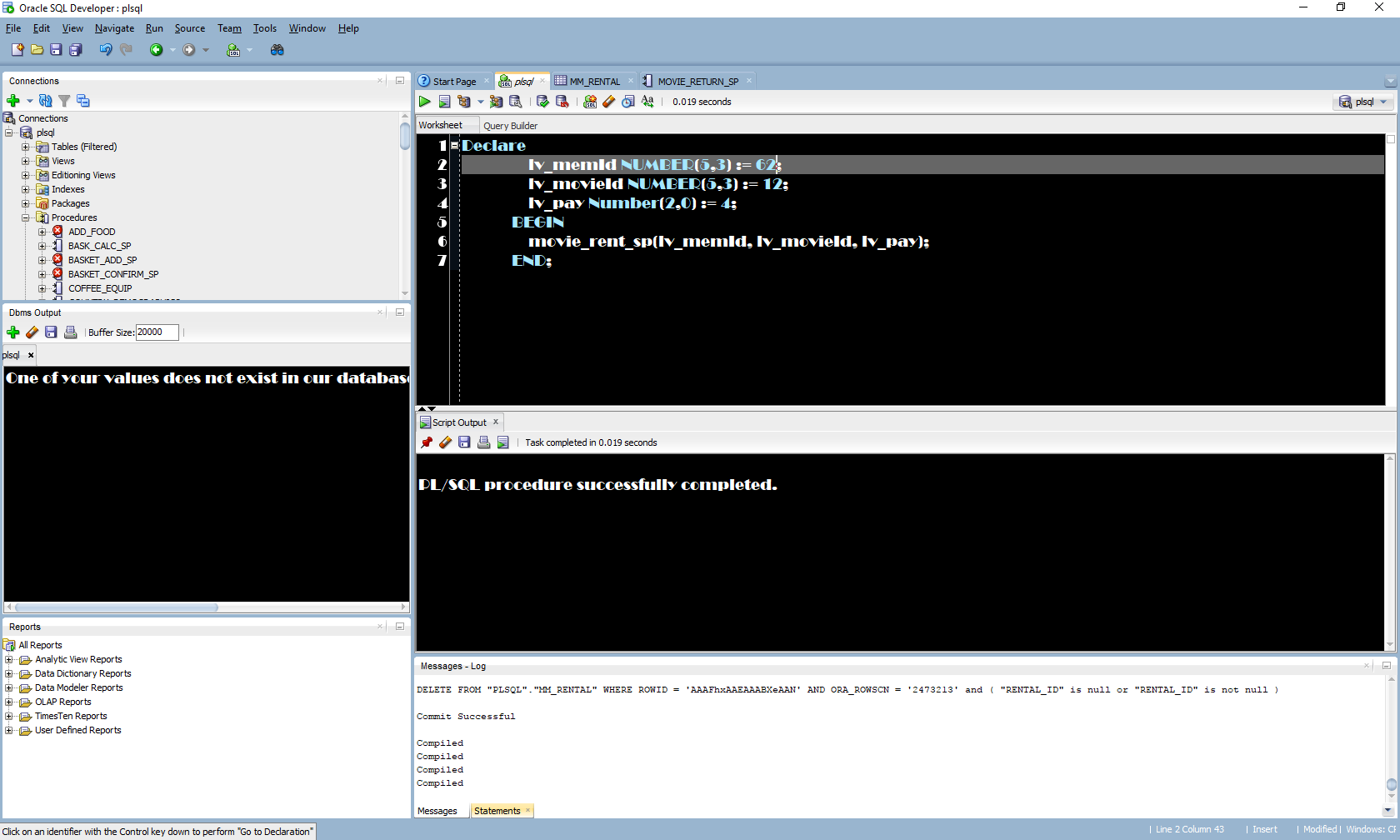
lv\_memId NUMBER(5,3) := 62;

lv\_movieId NUMBER(5,3) := 12;

lv\_pay Number(2,0) := 4;

BEGIN

movie\_rent\_sp(lv\_memId, lv\_movieId, lv\_pay);

 END;

--Part 1 Question 2

--PROCEDURE

CREATE OR REPLACE PROCEDURE MOVIE\_RETURN\_SP

(p\_memId IN NUMBER,

p\_movieId IN NUMBER)

AS

lv\_rental NUMBER(5,2);

lv\_qty NUMBER(5,2) := 1;

no\_info EXCEPTION;

PRAGMA EXCEPTION\_INIT(no\_info, -02291);

BEGIN

--Get the rental id

select rental\_id

into lv\_rental

from mm\_rental

where p\_memid = member\_id

AND p\_movieId = movie\_id;

--Check the movie back in

UPDATE mm\_rental

set checkin\_date = sysdate

where p\_memid = member\_id

AND p\_movieId = movie\_id

AND checkin\_date IS NULL;

--Update movie stock

Update mm\_movie

set MOVIE\_QTY = movie\_QTY + lv\_qty

WHERE p\_movieId = movie\_ID;

DBMS\_OUTPUT.PUT\_LINE('Movie rental id ' || lv\_rental || ' was checked in on ' || sysdate);

--Exceptions

exception

when no\_info then

DBMS\_OUTPUT.PUT\_LINE('One of your values does not exist in our database. Please review your values and try again.');

when NO\_DATA\_FOUND then

DBMS\_OUTPUT.PUT\_LINE('Sorry no data was found based on the given data.');

END MOVIE\_RETURN\_SP;

--Anon Block Successful

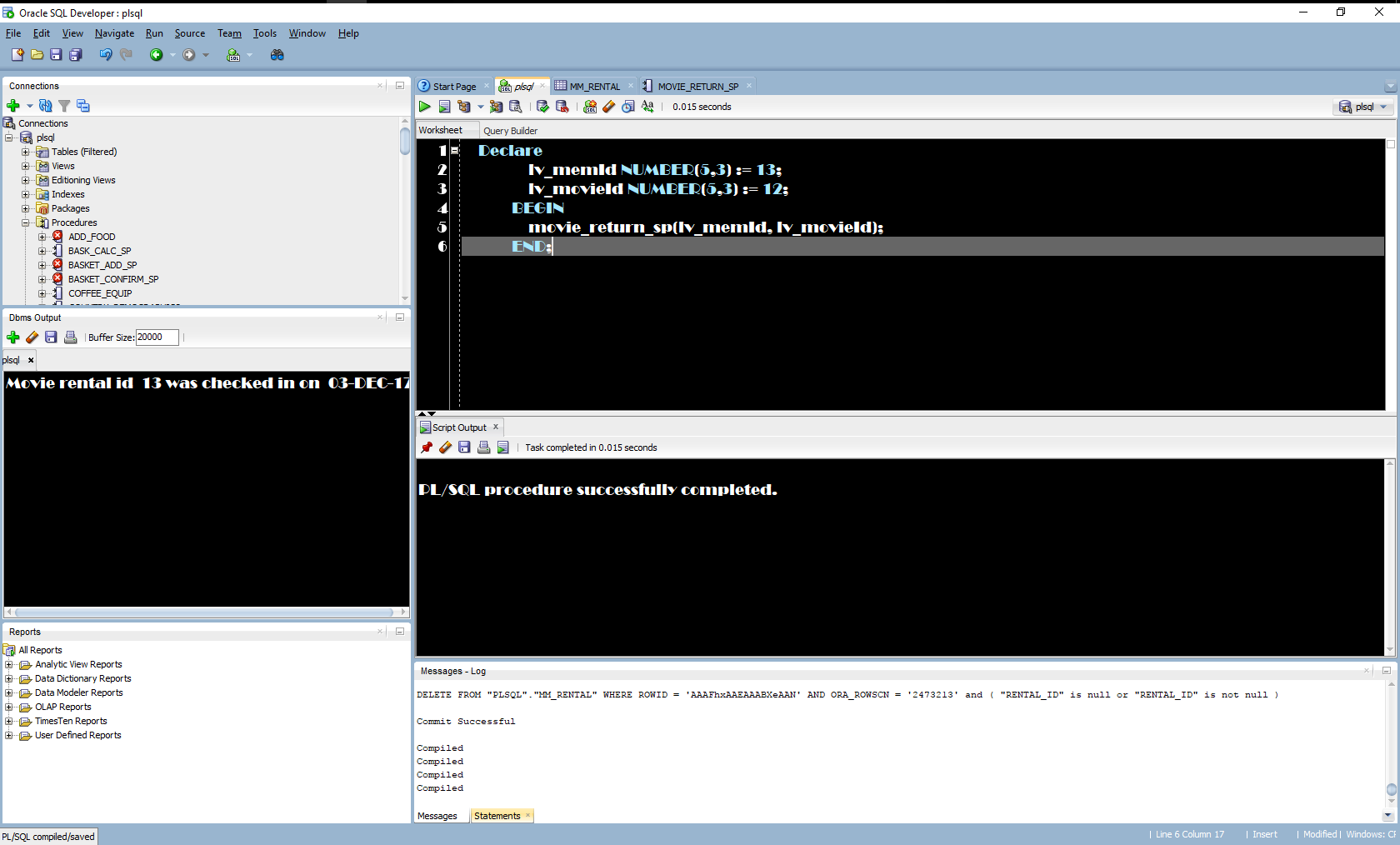
Declare

lv\_memId NUMBER(5,3) := 13;

lv\_movieId NUMBER(5,3) := 12;

BEGIN

movie\_return\_sp(lv\_memId, lv\_movieId);

 END;

--Anon Block Error

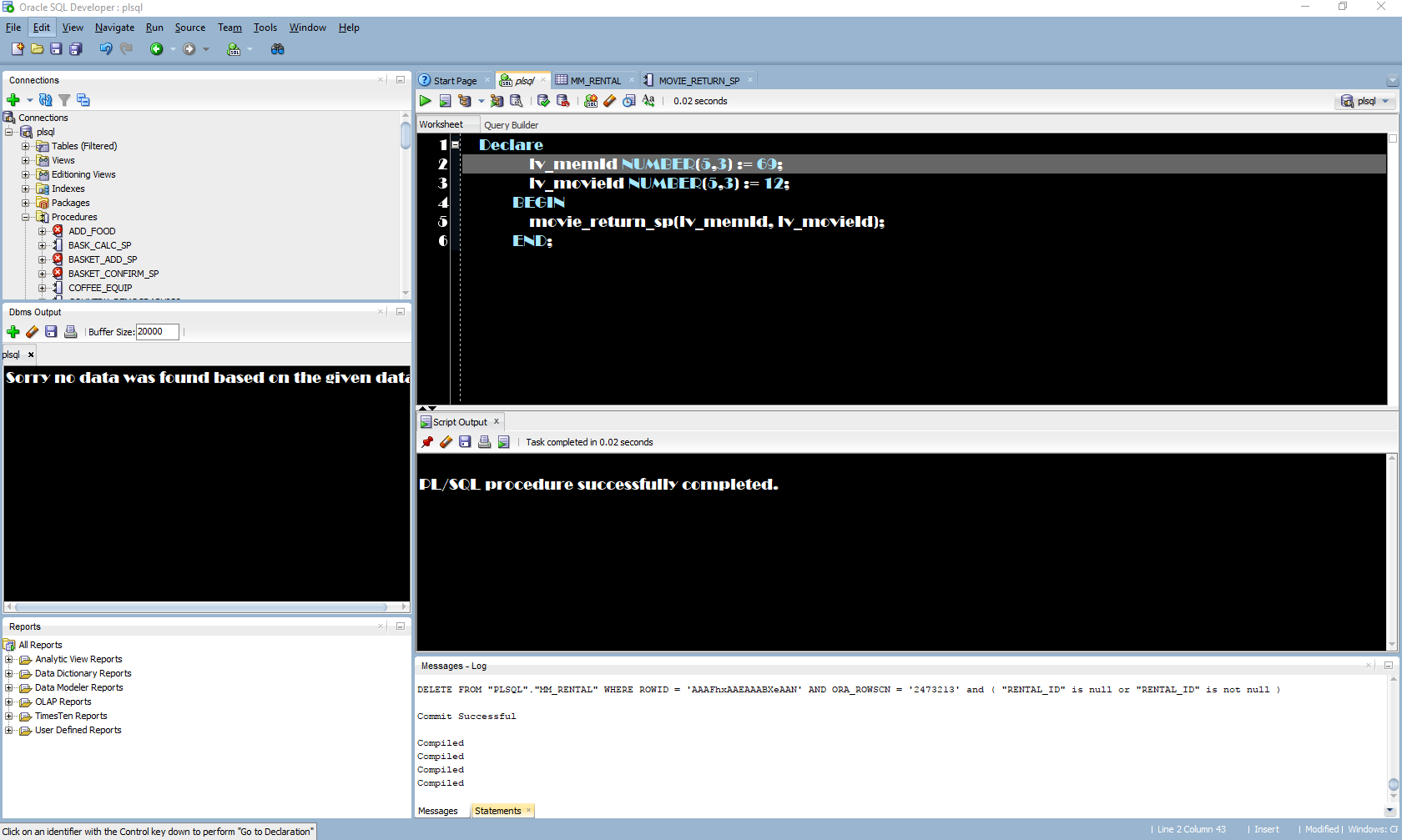
Declare

lv\_memId NUMBER(5,3) := 69;

lv\_movieId NUMBER(5,3) := 12;

BEGIN

movie\_return\_sp(lv\_memId, lv\_movieId);

 END;

--FUNCTION

create or replace FUNCTION MOVIE\_STOCK\_SF

(p\_movieId IN NUMBER)

RETURN VARCHAR2

AS

lv\_stock NUMBER(5,2);

lv\_title VARCHAR(255);

BEGIN

select MOVIE\_QTY, movie\_title

into lv\_stock, lv\_title

from mm\_movie

WHERE p\_movieId = movie\_ID;

if lv\_stock > 0 THEN

Return lv\_title || ' is available. There are ' || lv\_stock || ' left in stock.';

elsif lv\_stock = 0 THEN

Return lv\_title || ' is not in stock. Please check again later.';

Else

Return 'An error occurred finding the stcck. Please try again or contact customer service.';

END IF;

exception

when no\_data\_found then

RETURN 'That id does not match a movie in our database. Please review the id and try again.';

END MOVIE\_STOCK\_SF;

--Anon Block SUCCESSFUL

Declare

lv\_stock VARCHAR2(255);

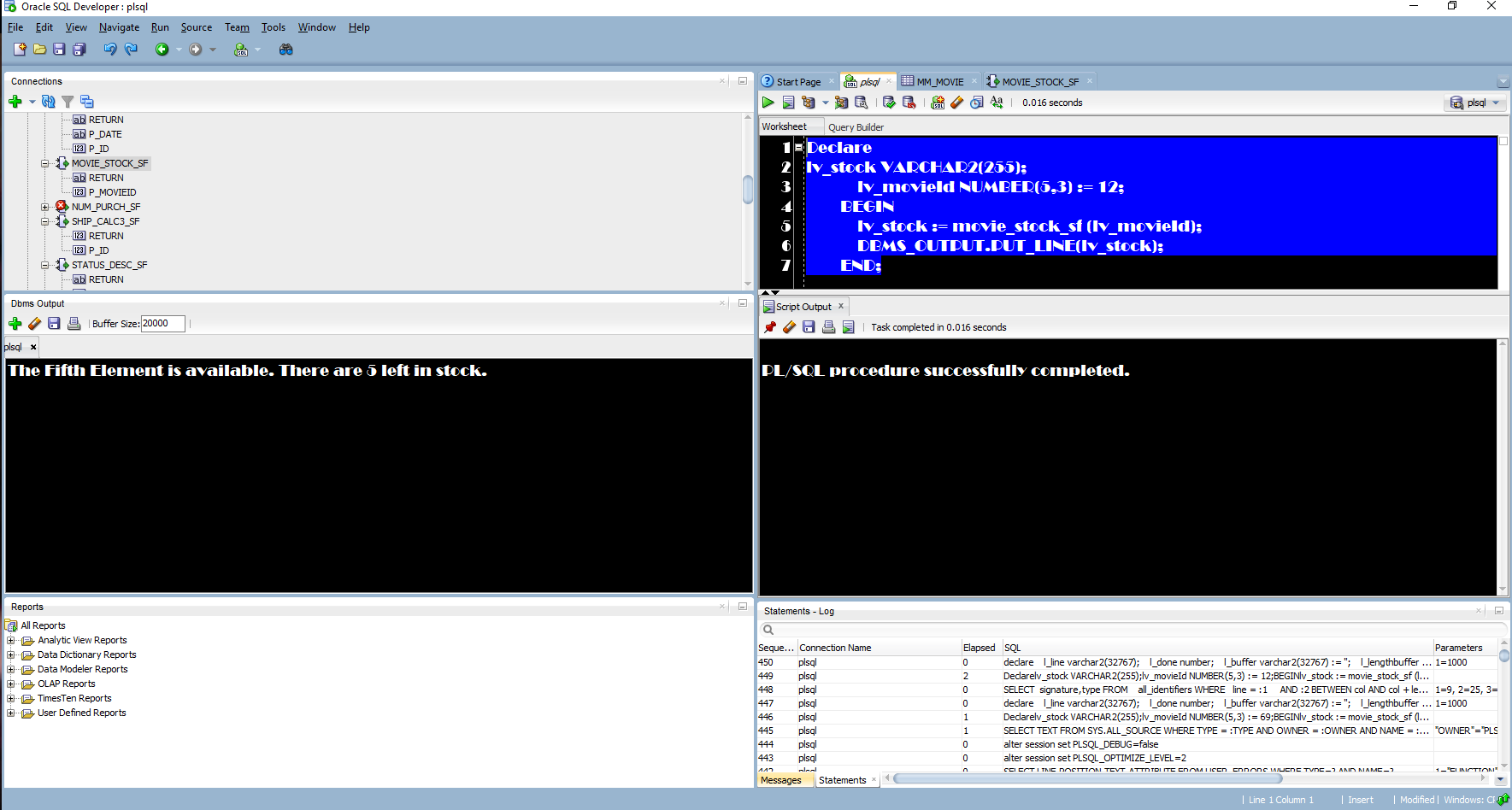
lv\_movieId NUMBER(5,3) := 12;

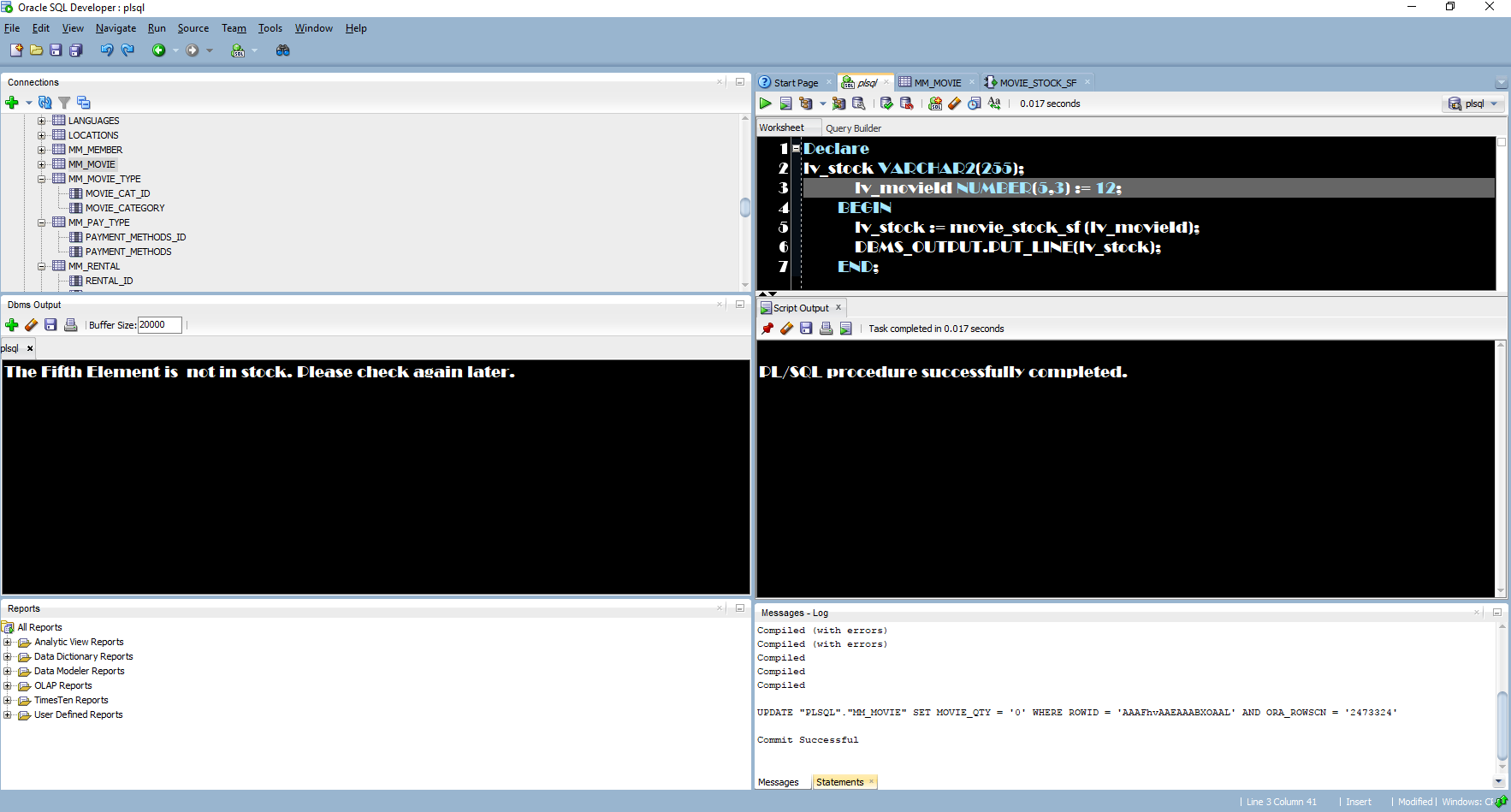
BEGIN

lv\_stock := movie\_stock\_sf (lv\_movieId);

DBMS\_OUTPUT.PUT\_LINE(lv\_stock);

END;



Out of stock

--Anon Block Error

Declare

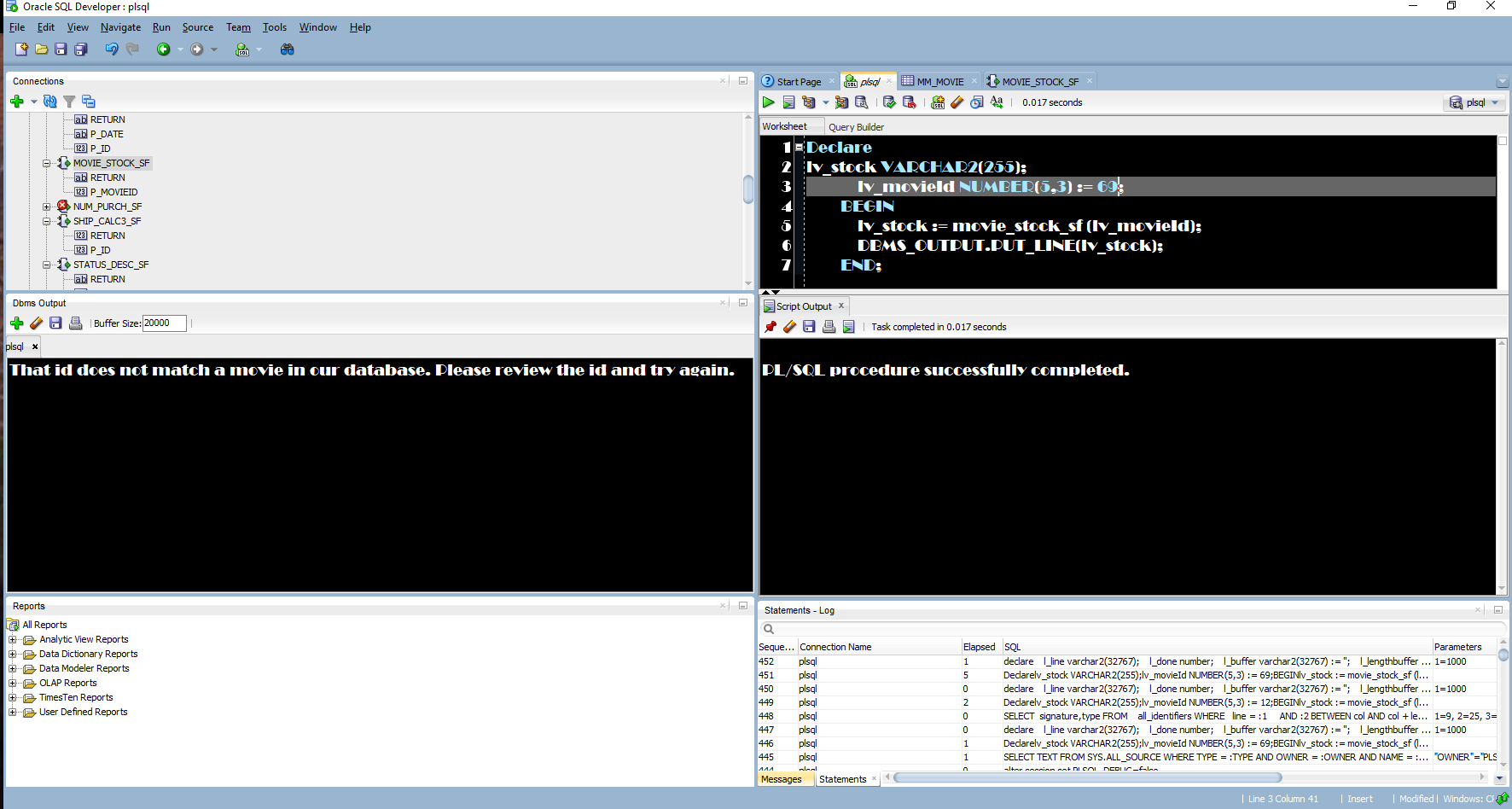
lv\_stock VARCHAR2(255);

lv\_movieId NUMBER(5,3) := 69;

BEGIN

lv\_stock := movie\_stock\_sf (lv\_movieId);

DBMS\_OUTPUT.PUT\_LINE(lv\_stock);

 END;

--Part 2 Question 2

--package create

create or replace PACKAGE MOVIE\_RENTALS\_PKG AS

FUNCTION MOVIE\_STOCK\_SF

(p\_movieId IN NUMBER)

RETURN VARCHAR2;

PROCEDURE MOVIE\_RENT\_SP

(p\_memId in OUT NUMBER,

p\_movieId in OUT Number,

p\_pay in OUT Number);

PROCEDURE MOVIE\_RETURN\_SP

(p\_memId IN NUMBER,

p\_movieId IN NUMBER);

END MOVIE\_RENTALS\_PKG;

--Package BODY

CREATE OR REPLACE

PACKAGE BODY MOVIE\_RENTALS\_PKG AS

FUNCTION MOVIE\_STOCK\_SF

(p\_movieId IN NUMBER)

RETURN VARCHAR2

AS

lv\_stock NUMBER(5,2);

lv\_title VARCHAR(255);

BEGIN

select MOVIE\_QTY, movie\_title

into lv\_stock, lv\_title

from mm\_movie

WHERE p\_movieId = movie\_ID;

if lv\_stock > 0 THEN

Return lv\_title || ' is available. There are ' || lv\_stock || ' left in stock.';

elsif lv\_stock = 0 THEN

Return lv\_title || ' is not in stock. Please check again later.';

Else

Return 'An error occurred finding the stcck. Please try again or contact customer service.';

END IF;

exception

when no\_data\_found then

RETURN 'That id does not match a movie in our database. Please review the id and try again.';

END MOVIE\_STOCK\_SF;

PROCEDURE MOVIE\_RENT\_SP

(p\_memId in OUT NUMBER,

p\_movieId in OUT Number,

p\_pay in OUT Number)

AS

lv\_qty NUMBER(5,2) := 1;

lv\_stock NUMBER(5,2);

no\_info EXCEPTION;

PRAGMA EXCEPTION\_INIT(no\_info, -02291);

BEGIN

--Create new order

Insert into mm\_rental(rental\_id,member\_id, movie\_ID, PAYMENT\_METHODS\_ID)

VALUES (MM\_RENTAL\_SEQ.NEXTVAL,p\_memid, p\_movieId, p\_pay);

--Update Stock

Update mm\_movie

set MOVIE\_QTY = movie\_QTY - lv\_qty

WHERE p\_movieId = movie\_ID

AND movie\_qty > 0;

--Put stock into variable to output

select MOVIE\_QTY

into lv\_stock

from mm\_movie

WHERE p\_movieId = movie\_ID

AND movie\_qty > 0;

--Output everything

DBMS\_OUTPUT.PUT\_LINE('New Order Report Created On: ' || sysdate);

DBMS\_OUTPUT.PUT\_LINE('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

DBMS\_OUTPUT.PUT\_LINE('Rental Id: ' || MM\_RENTAL\_SEQ.NEXTVAL);

DBMS\_OUTPUT.PUT\_LINE('Member Id: ' || p\_memId);

DBMS\_OUTPUT.PUT\_LINE('Movie Id: ' || p\_movieId);

DBMS\_OUTPUT.PUT\_LINE('Payment Type: ' || p\_pay);

DBMS\_OUTPUT.PUT\_LINE('Movie Stock: ' || lv\_stock);

--Exceptions

exception

when no\_info then

DBMS\_OUTPUT.PUT\_LINE('One of your values does not exist in our database. Please review your values and try again.');

when NO\_DATA\_FOUND then

--Update Stock

Insert into mm\_rental(rental\_id,member\_id, movie\_ID, CHECKOUT\_DATE, PAYMENT\_METHODS\_ID, notes)

VALUES (MM\_RENTAL\_SEQ.NEXTVAL,p\_memid, p\_movieId, NULL, NULL, 'Order not completed. Movie not in stock. Contact customer.');

DBMS\_OUTPUT.PUT\_LINE('Sorry, that movie is currently not in stock.');

END MOVIE\_RENT\_SP;

PROCEDURE MOVIE\_RETURN\_SP

(p\_memId IN NUMBER,

p\_movieId IN NUMBER)

AS

lv\_rental NUMBER(5,2);

lv\_qty NUMBER(5,2) := 1;

no\_info EXCEPTION;

PRAGMA EXCEPTION\_INIT(no\_info, -02291);

BEGIN

--Get the rental id

select rental\_id

into lv\_rental

from mm\_rental

where p\_memid = member\_id

AND p\_movieId = movie\_id;

--Check the movie back in

UPDATE mm\_rental

set checkin\_date = sysdate

where p\_memid = member\_id

AND p\_movieId = movie\_id

AND checkin\_date IS NULL;

--Update movie stock

Update mm\_movie

set MOVIE\_QTY = movie\_QTY + lv\_qty

WHERE p\_movieId = movie\_ID;

DBMS\_OUTPUT.PUT\_LINE('Movie rental id ' || lv\_rental || ' was checked in on ' || sysdate);

--Exceptions

exception

when no\_info then

DBMS\_OUTPUT.PUT\_LINE('One of your values does not exist in our database. Please review your values and try again.');

when NO\_DATA\_FOUND then

DBMS\_OUTPUT.PUT\_LINE('Sorry no data was found based on the given data.');

END MOVIE\_RETURN\_SP;

END MOVIE\_RENTALS\_PKG;

--Anon Block id:6

Declare

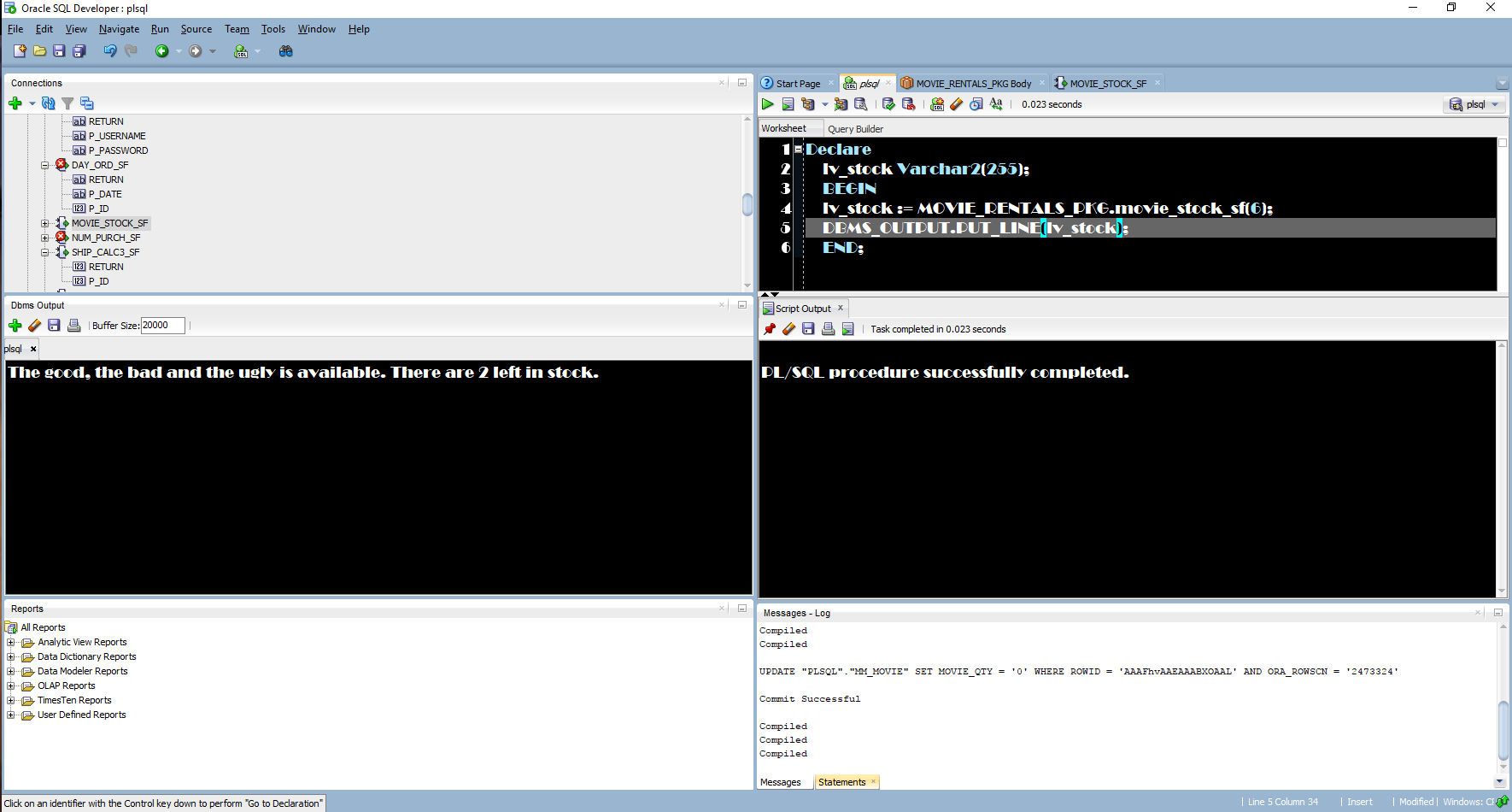
lv\_stock Varchar2(255);

BEGIN

lv\_stock := MOVIE\_RENTALS\_PKG.movie\_stock\_sf(6);

DBMS\_OUTPUT.PUT\_LINE(lv\_stock);

END;



--Anon block id:7

Declare

lv\_stock Varchar2(255);

BEGIN

lv\_stock := MOVIE\_RENTALS\_PKG.movie\_stock\_sf(7);

DBMS\_OUTPUT.PUT\_LINE(lv\_stock);

END;

--Part 3 Question 1

--PACKAGE create

create or replace PACKAGE MOVIE\_RENTALS\_PKG AS

FUNCTION MOVIE\_STOCK\_SF

(p\_movieId IN NUMBER)

RETURN VARCHAR2;

FUNCTION MOVIE\_STOCK\_SF

(p\_movieId IN VARCHAR2)

RETURN VARCHAR2;

PROCEDURE MOVIE\_RENT\_SP

(p\_memId in OUT NUMBER,

p\_movieId in OUT Number,

p\_pay in OUT Number);

PROCEDURE MOVIE\_RETURN\_SP

(p\_memId IN NUMBER,

p\_movieId IN NUMBER);

END MOVIE\_RENTALS\_PKG;

--Package BODY

create or replace PACKAGE BODY MOVIE\_RENTALS\_PKG AS

FUNCTION MOVIE\_STOCK\_SF

(p\_movieId IN NUMBER)

RETURN VARCHAR2

AS

lv\_stock NUMBER(5,2);

lv\_title VARCHAR(255);

BEGIN

select MOVIE\_QTY, movie\_title

into lv\_stock, lv\_title

from mm\_movie

WHERE p\_movieId = movie\_ID;

if lv\_stock > 0 THEN

Return lv\_title || ' is available. There are ' || lv\_stock || ' left in stock.';

elsif lv\_stock = 0 THEN

Return lv\_title || ' is not in stock. Please check again later.';

Else

Return 'An error occurred finding the stcck. Please try again or contact customer service.';

END IF;

exception

when no\_data\_found then

RETURN 'That id does not match a movie in our database. Please review the id and try again.';

END MOVIE\_STOCK\_SF;

FUNCTION MOVIE\_STOCK\_SF

(p\_movieId IN VARCHAR2)

RETURN VARCHAR2

AS

lv\_stock NUMBER(5,2);

lv\_title VARCHAR(255);

BEGIN

select MOVIE\_QTY, movie\_title

into lv\_stock, lv\_title

from mm\_movie

WHERE p\_movieId = movie\_title;

if lv\_stock > 0 THEN

Return lv\_title || ' is available. There are ' || lv\_stock || ' left in stock.';

elsif lv\_stock = 0 THEN

Return lv\_title || ' is not in stock. Please check again later.';

Else

Return 'An error occurred finding the stcck. Please try again or contact customer service.';

END IF;

exception

when no\_data\_found then

RETURN 'That title does not match a movie in our database. Please review the id and try again.';

END MOVIE\_STOCK\_SF;

PROCEDURE MOVIE\_RENT\_SP

(p\_memId in OUT NUMBER,

p\_movieId in OUT Number,

p\_pay in OUT Number)

AS

lv\_qty NUMBER(5,2) := 1;

lv\_stock NUMBER(5,2);

no\_info EXCEPTION;

PRAGMA EXCEPTION\_INIT(no\_info, -02291);

BEGIN

--Create new order

Insert into mm\_rental(rental\_id,member\_id, movie\_ID, PAYMENT\_METHODS\_ID)

VALUES (MM\_RENTAL\_SEQ.NEXTVAL,p\_memid, p\_movieId, p\_pay);

--Update Stock

Update mm\_movie

set MOVIE\_QTY = movie\_QTY - lv\_qty

WHERE p\_movieId = movie\_ID

AND movie\_qty > 0;

--Put stock into variable to output

select MOVIE\_QTY

into lv\_stock

from mm\_movie

WHERE p\_movieId = movie\_ID

AND movie\_qty > 0;

--Output everything

DBMS\_OUTPUT.PUT\_LINE('New Order Report Created On: ' || sysdate);

DBMS\_OUTPUT.PUT\_LINE('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

DBMS\_OUTPUT.PUT\_LINE('Rental Id: ' || MM\_RENTAL\_SEQ.NEXTVAL);

DBMS\_OUTPUT.PUT\_LINE('Member Id: ' || p\_memId);

DBMS\_OUTPUT.PUT\_LINE('Movie Id: ' || p\_movieId);

DBMS\_OUTPUT.PUT\_LINE('Payment Type: ' || p\_pay);

DBMS\_OUTPUT.PUT\_LINE('Movie Stock: ' || lv\_stock);

--Exceptions

exception

when no\_info then

DBMS\_OUTPUT.PUT\_LINE('One of your values does not exist in our database. Please review your values and try again.');

when NO\_DATA\_FOUND then

--Update Stock

Insert into mm\_rental(rental\_id,member\_id, movie\_ID, CHECKOUT\_DATE, PAYMENT\_METHODS\_ID, notes)

VALUES (MM\_RENTAL\_SEQ.NEXTVAL,p\_memid, p\_movieId, NULL, NULL, 'Order not completed. Movie not in stock. Contact customer.');

DBMS\_OUTPUT.PUT\_LINE('Sorry, that movie is currently not in stock.');

END MOVIE\_RENT\_SP;

PROCEDURE MOVIE\_RETURN\_SP

(p\_memId IN NUMBER,

p\_movieId IN NUMBER)

AS

lv\_rental NUMBER(5,2);

lv\_qty NUMBER(5,2) := 1;

no\_info EXCEPTION;

PRAGMA EXCEPTION\_INIT(no\_info, -02291);

BEGIN

--Get the rental id

select rental\_id

into lv\_rental

from mm\_rental

where p\_memid = member\_id

AND p\_movieId = movie\_id;

--Check the movie back in

UPDATE mm\_rental

set checkin\_date = sysdate

where p\_memid = member\_id

AND p\_movieId = movie\_id

AND checkin\_date IS NULL;

--Update movie stock

Update mm\_movie

set MOVIE\_QTY = movie\_QTY + lv\_qty

WHERE p\_movieId = movie\_ID;

DBMS\_OUTPUT.PUT\_LINE('Movie rental id ' || lv\_rental || ' was checked in on ' || sysdate);

--Exceptions

exception

when no\_info then

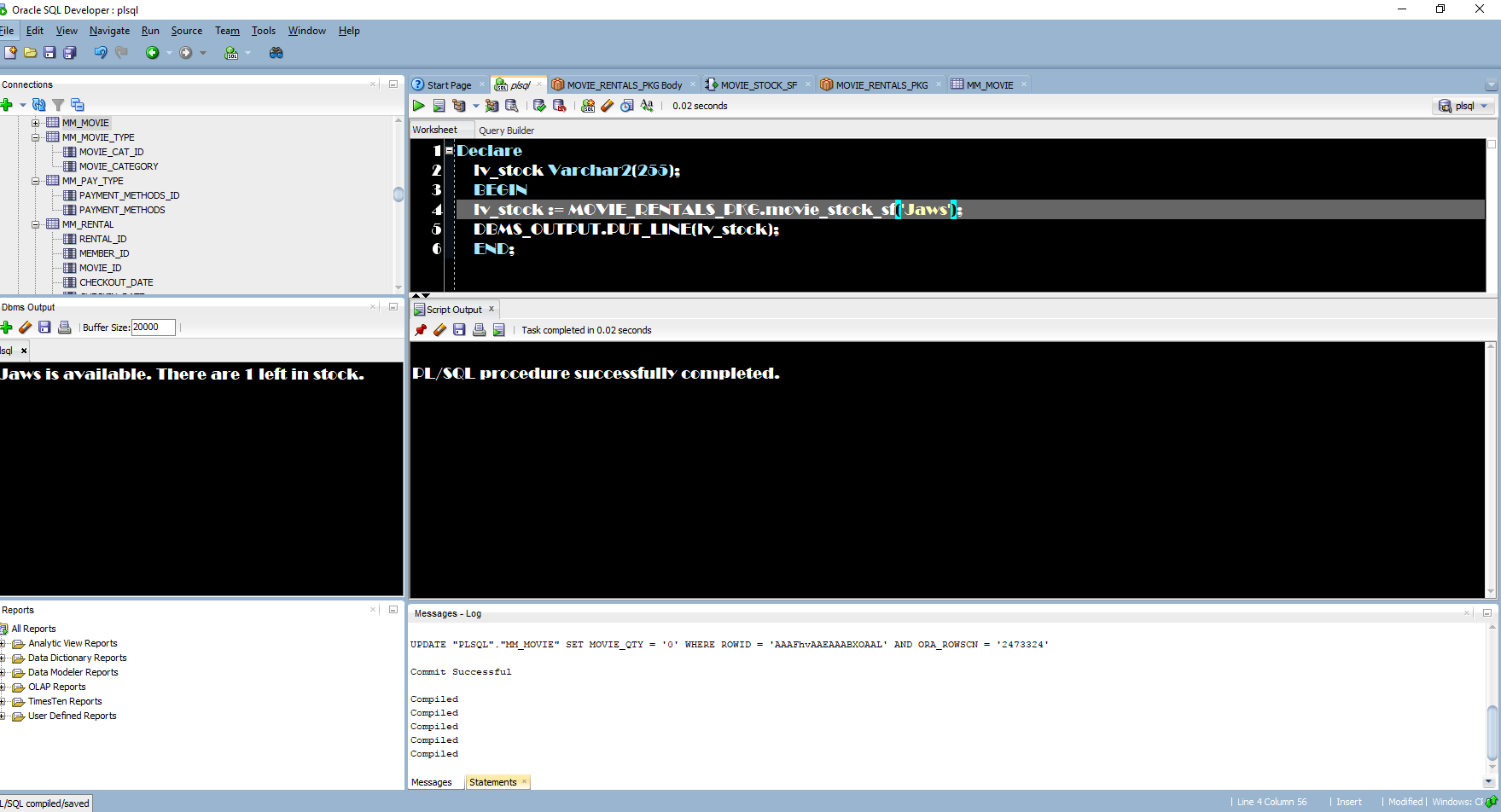
DBMS\_OUTPUT.PUT\_LINE('One of your values does not exist in our database. Please review your values and try again.');

when NO\_DATA\_FOUND then

DBMS\_OUTPUT.PUT\_LINE('Sorry no data was found based on the given data.');

END MOVIE\_RETURN\_SP;

END MOVIE\_RENTALS\_PKG;

 --anon block successful

Declare

lv\_stock Varchar2(255);

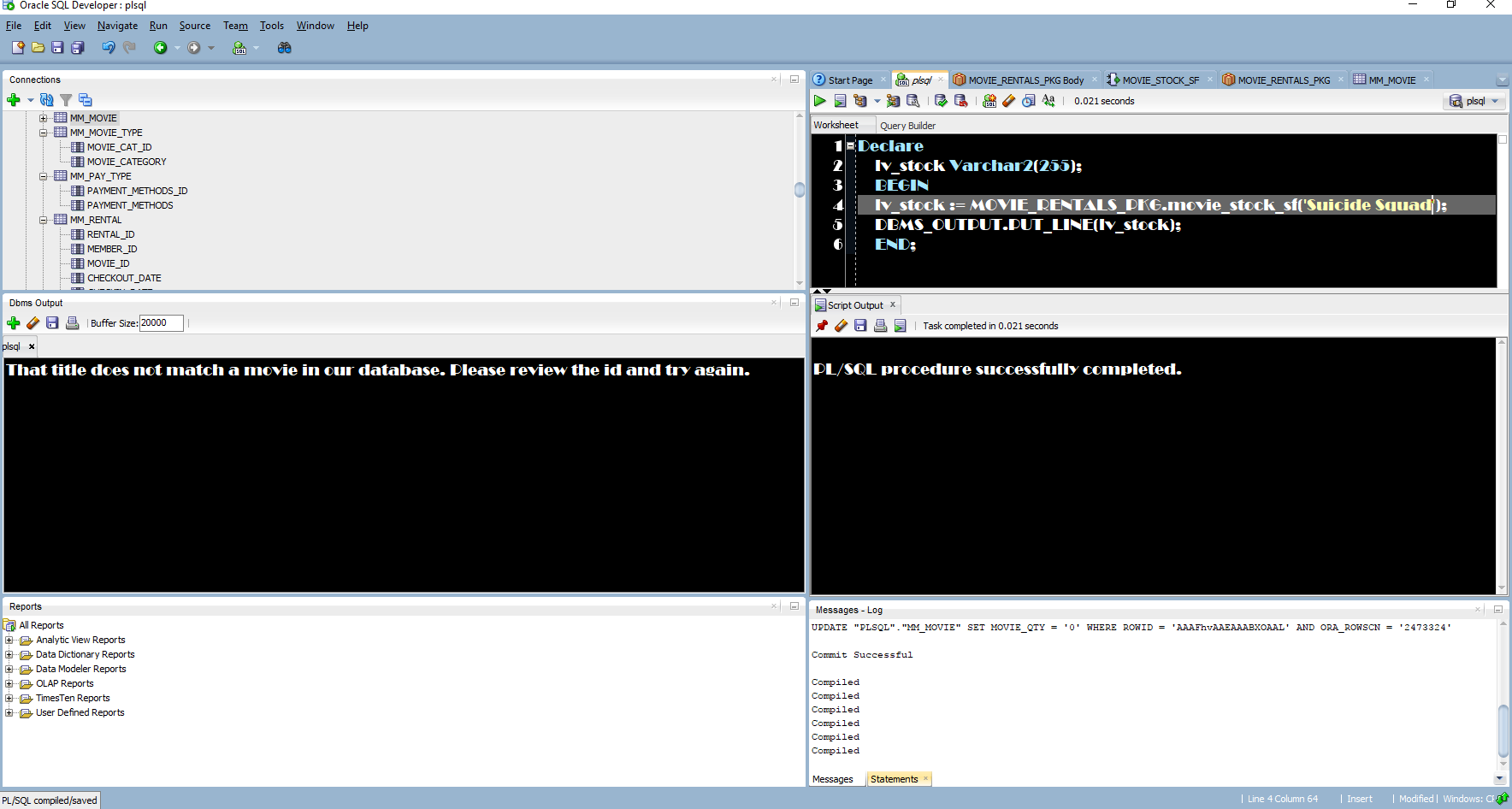
BEGIN

lv\_stock := MOVIE\_RENTALS\_PKG.movie\_stock\_sf('Jaws');

DBMS\_OUTPUT.PUT\_LINE(lv\_stock);

END;

--anon block error



Declare

lv\_stock Varchar2(255);

BEGIN

lv\_stock := MOVIE\_RENTALS\_PKG.movie\_stock\_sf('Saw');

DBMS\_OUTPUT.PUT\_LINE(lv\_stock);

END;

--Part 3 Question 2

--Function

CREATE OR REPLACE FUNCTION MOVIE\_SUGGESTION\_SF

(p\_category IN VARCHAR2)

RETURN VARCHAR2

AS

no\_value exception;

PRAGMA EXCEPTION\_INIT(no\_value, -06503);

CURSOR category IS

SELECT movie\_category, movie\_title

FROM MM\_MOVIE NATURAL JOIN MM\_MOVIE\_TYPE

WHERE UPPER(p\_category) = UPPER(movie\_category);

lv\_name category%ROWTYPE;

BEGIN

open category;

LOOP

fetch category into lv\_name;

IF category%Rowcount >= 1 THEN

dbms\_output.put\_line(lv\_name.movie\_title);

ELSE DBMS\_OUTPUT.PUT\_LINE('Sorry, our system could not find a movie title within that category.');

END IF;

exit when category%NOTFOUND;

END LOOP;

close category;

exception

when no\_value then

DBMS\_OUTPUT.PUT\_LINE('Sorry, our system could not find a movie title within that category.');

when no\_data\_found then

DBMS\_OUTPUT.PUT\_LINE('Sorry, our system could not find a movie title within that category.');

END MOVIE\_SUGGESTION\_SF;

--Anon Block

DECLARE

lv\_category VARCHAR2(255) :='&Category';

lv\_titles Varchar2(255);

BEGIN

DBMS\_OUTPUT.Put\_Line('Movies Titles from that cateogry:');

DBMS\_OUTPUT.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

lv\_titles := MOVIE\_SUGGESTION\_SF(lv\_category);

END;

